Name: \_\_Replace with your name(s)\_\_\_\_

EID: \_\_Replace with your EID(s)\_\_\_\_\_

Semester: Fall 2024

Course: ECE445L

1. ***Requirements Document:***
2. I have completed the Project Requirements Document at the end of this lab document (Type yes if true): \_\_\_\_\_

B) ***Objectives*:**

1. In a few sentences, summarize the goal of this lab.

C) ***Hardware Design*:**

1. Schematic showing any additional hardware add to the robot.

D) ***Software Design Deliverables:***

1. I have pushed my code to GitHub for grading (Type yes if true): \_\_\_\_\_
2. Briefly describe the system design. Alternatively, create a system call graph.

E) ***Measurement Data:***

1. Deliverable 1: Measure the voltage drop across a motor when on (like Figure 8.6)
2. Deliverable 2: Graph of motor power versus PWM duty cycle (like Figure 8.11). Explain why or why not the motor behaves like a simple resistive load.
3. Deliverable 3: Scope capture of tachometer output (like Figure 8.9)
4. Deliverable 4: Motor speed versus applied power graph (like Figure 8.12)
5. Deliverable 5: Speed versus time graphs of 4 PI configurations (like Figure 8.10)
6. Deliverable 6: CPU utilization to run the PI controller
7. Deliverable 7: Rise time, settling time, and peak overshoot ratio for the three stable settings
8. Deliverable 8: Measure the total current to RSLK at full speed.

F) ***Analysis and Discussion Questions:***

1. What is torque? What are its units?
2. You started with an incremental controller because it is simple and stable. Compare and contrast the advantages of the integral versus proportional-integral controller?
3. Why do we need to use a PI loop to control the brushed DC motor? Would we have needed a PI loop to control a servo, a BLDC, or a stepper? What are the differences?
4. Explain how the data collected in deliverables 1 and 2 are important for choosing a motor drive interface chip. Which driver chip DRV8847 or L293D better satisfies these parameters?
5. How did you tune your PI controller? What was your process? Is there a better way?
6. Compare and contrast these two implementations of the I term

LeftI = LeftI + (Ki1 \* LeftE)/Ki2;

and

LeftBuf[I] = LeftE; I = (I+1)%100;

LeftSum = 0; for(int i=0; i<100; i++){LeftSum += LeftBuf[i];}

AveLeftE = LeftSum/100;

LeftI = (Ki1 \* AveLeftE)/Ki2;

F) ***Project Requirements Document:***

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

The objectives of this project are to control the two brushed DC motors on the RSLK robot. The two motors should spin at the same constant speed, and the operator can specify the desired set point. Educationally, students are learning how to interface a DC motor, how to measure speed using input capture, and how to implement a digital controller running in the background.

1.2. Process: How will the project be developed?

The project will be developed using the TM4C LaunchPad attached to the RSLK robot. The user will use the UART to specify the desired speed of the motor as well as the “K” terms for the Proportional and Integral components of the PI controller. There is a solderless breadboard that can be used if the tachometer needs digital or analog filtering. RSLK is powered with 6 NiMH rechargeable batteries. A hardware/software interface will be designed that allows software to control the DC motor. There are four hardware/software modules: motor output, tachometer input, digital controller, and CLI user interface using UART. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

ECE445L students are the engineers, and the TA is the client. Student 1 will design, build and test the motor controller output. Student 2 will design, build and test the tachometer input. Student 3 will design, build and test the digital controller. Student 4 will design, build and test the CLI interface. All students will work to collect performance measurements and tuning the controller. *(Note to students: you are expected to make minor modifications to this document to clarify exactly what you plan to build. Students are allowed to divide responsibilities of the project however they wish, but, at the time of demonstration, all students are expected to understand all aspects of the design.)*

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the microcontroller LaunchPad and the RSLK robot. The circuit diagrams for the robot are in the **RSLK2\_Description.docx** file.

1.5. Terminology: Define terms used in the document.

For the terms Proportional-Integral (PI) controller, PWM, board support package, back EMF, torque, time constant, and hysteresis, see textbook for definitions.

1.6. Security: How will intellectual property be managed?

The system may include software from the class website and from the book. No software written for this project may be transmitted, viewed, or communicated with any other ECE445L student past, present, or future (other than the lab partner of course). **It is the responsibility of each team to keep its EE445L lab solutions secure.**

2. Function Description

2.1. Functionality: What will the system do precisely?

The CLI will provide the following capabilities:

* Entry to set the desired motor speed
* Two entries to set the Kp1 (multiplier) and Kp2 (divider) values for the Proportional term
* Two entries to set the Kp1 (multiplier) and Kp2 (divider) values for the Integral term
* The CLI will have the capability to display the modified value after it has been entered
* The CLI will have the capability to handle the BS and DEL keys, where the character entered is deleted from the input buffer and the display refreshed with the reentered value
* The CLI will ignore CR only entries and keep the previous values (Figure a)

A picture containing graphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure a Figure b

* CTRL-C or c: Cancel the entry sequence and redisplay all the current motor control register values: New\_Speed, Kp1, Kp2, Ki1and Ki2 (Figure b)
* CTRL-S or s: Display status (Figure c), once or continuously. In this mode the following must be displayed:
* Xstar & RPM Values
* PI Error Values (E)
* PWM Duty Cycle Control Values (U)

Table

Description automatically generated with medium confidence Table

Description automatically generated with medium confidence

Redo

Figure c Figure d

* d: Display two seconds of speed versus time data collected immediately after the set-point has changed (Figure d).

*(Note to students: feel free to change the look and feel of the CLI. However, edit this document to describe how it actually works.)*

The PI control equation may be in this form:

LeftE = Xstar-Tachometer\_GetLeftrpm(); // 0.1 RPM

LeftP = (Kp1 \* LeftE)/Kp2; // Proportional term

LeftI = LeftI + (Ki1 \* LeftE)/Ki2; // SUM(KiDt)

if(LeftI < MOTORMIN) LeftI = MOTORMIN; // anti reset wind-up

if(LeftI > MOTORMAX) LeftI = MOTORMAX; // 2 to 39998

LeftU = LeftP + LeftI; // Calculate the actuator value

if(LeftU < MOTORMIN) LeftU=MOTORMIN; // Minimum PWM output

if(LeftU > MOTORMAX) LeftU=MOTORMAX; // 2 to 39998

Motor\_Forward(LeftU,RightU); // Send to PWM

There will be separate controllers for left and right, but one set of {Kp1, Kp2, Ki1, Ki2} parameters. Note that the proportional and integral control values in the PI loop each have two sub-values, one is the multiplicand, and the other is the divisor. To prevent underflow, we always multiply first and then divide. This provides for non-integer values for the integral and proportional gain terms. Note the P term may go negative, but the I and U values will remain between 2 and 39998.

The motor speed should start out at zero RPM. Once the desired motor speed is entered the motor should start. *(Note to students: feel free to change how the set point is established, and feel free to increase or decrease the maximum speed in accordance to how it actually works.)*

The motor power and desired speed should be set to 0 if any of the bump switches are activated.

Both the desired and actual speeds should be displayed on the OLED. (Note to students: feel free to specify exactly how the data is displayed. For example, you could but do not have to add numerical outputs).

2.2. Scope: List the phases and what will be delivered in each phase.

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the LaunchPad and RSLK robot will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the system must employ a PI controller running in the background. There should be a clear and obvious abstraction, separating the state estimator, user interface, the controller and the actuator output. Backward jumps in the ISR are not allowed. Third, all software will be judged according to style guidelines. Software must follow the style described in Section 3.3 of the book. *(Note to students: you may edit this sentence to define a different style format)*. There are steady state and transient quantitative measures. In steady state, the average speed error at a desired speed of 60 RPM will be measured. The average error should be less than 5 RPM. After a change in set-point, you will measure rise time, settling time, and peak overshoot ratio. Lastly, you will measure power supply current to run the system. There is no particular need to minimize controller error, step response, or system current in this system.

2.5. Usability: Describe the interfaces. Be quantitative if possible. Describe the how the CLI will control the motor.

2.6. Safety: Explain any safety requirements and how they will be measured.

We expect each motor current to be less than 120 mA. Please place a current probe on the 7.2V power during initial testing to make sure your system is behaving correctly. However, under heavy friction this current could be 2 times higher. Therefore, please run the motor unloaded. Connecting or disconnecting wires on the robot while power is applied will damage the microcontroller. Operating the circuit without a snubber diode will also damage the microcontroller.

3. Deliverables

3.1. Reports: How will the system be described?

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

3.2. Audits: How will the clients evaluate progress?

The preparation is due at the beginning of the lab period on the date listed in the syllabus.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

There are three deliverables: preparation, demonstration, and report. *(Note to students: you should remove all notes to students in your final requirements document).*